

CLOUD COMPUTING CAT-2 PROJECT

M.ROHAN	1934034
A.SARAVANA PRABHU	1934039
D.KISHORE KUMAR	1934017
G.DHINESH	1934007

PROBLEM STATEMENT:

A project for people who want to explore comic and manga culture. In this site users can view our favorite manga and comic covers, Even buy a comic they wish. Goal of this project is to make lightNovel readers and novel readers to give a taste of manga culture.

PROPOSED SOLUTION:

Our attempt is to create a comic book store to increase the readers group worldwide using cloud services Docker containers and kubernetes.

Dockerfile

```
👉 Dockerfile X
024pricing-plans > dist > 🚀 Dockerfile > ⚙ FROM
  1   FROM nginx:alpine
  2   COPY . /usr/share/nginx/html
```

Building Docker image:

```
D:\sem\5\DCC\project\024pricing-plans\dist>docker build -t pricing:v1 .
[+] Building 4.6s (7/7) FINISHED
=> [internal] load build definition from Dockerfile          0.4s
=> => transferring dockerfile: 84B                           0.0s
=> [internal] load .dockerignore                            0.4s
=> => transferring context: 2B                           0.0s
=> [internal] load metadata for docker.io/library/nginx:alpine   3.9s
=> [internal] load build context                          0.3s
=> => transferring context: 4.16kB                      0.0s
=> CACHED [1/2] FROM docker.io/library/nginx@sha256:da9c94bec1da829ebd52431a84502ec471c8e548ffb2cedbf3626 0.0s
=> [2/2] COPY . /usr/share/nginx/html                     0.0s
=> exporting to image                                     0.0s
=> => exporting layers                                    0.0s
=> => writing image sha256:88d3f398cd6630065eea8d44d2db7711934eee54fc5267272c2f61214668be3c 0.0s
=> => naming to docker.io/library/pricing:v1            0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

Running image:

```
C:\Users\mural\Downloads\web\comic html\WEB DEV>docker run -d -p 80:80 nerd:v1
75d9f0662ac60e74f3b9f6dd2617df706ac09d89748aa4b11bc58a6908e56a51
```

Image caching:

nerd	IN USE	v1	ae9ceb58b9ba	2 minutes ago	363.69 MB
------	--------	----	--------------	---------------	-----------

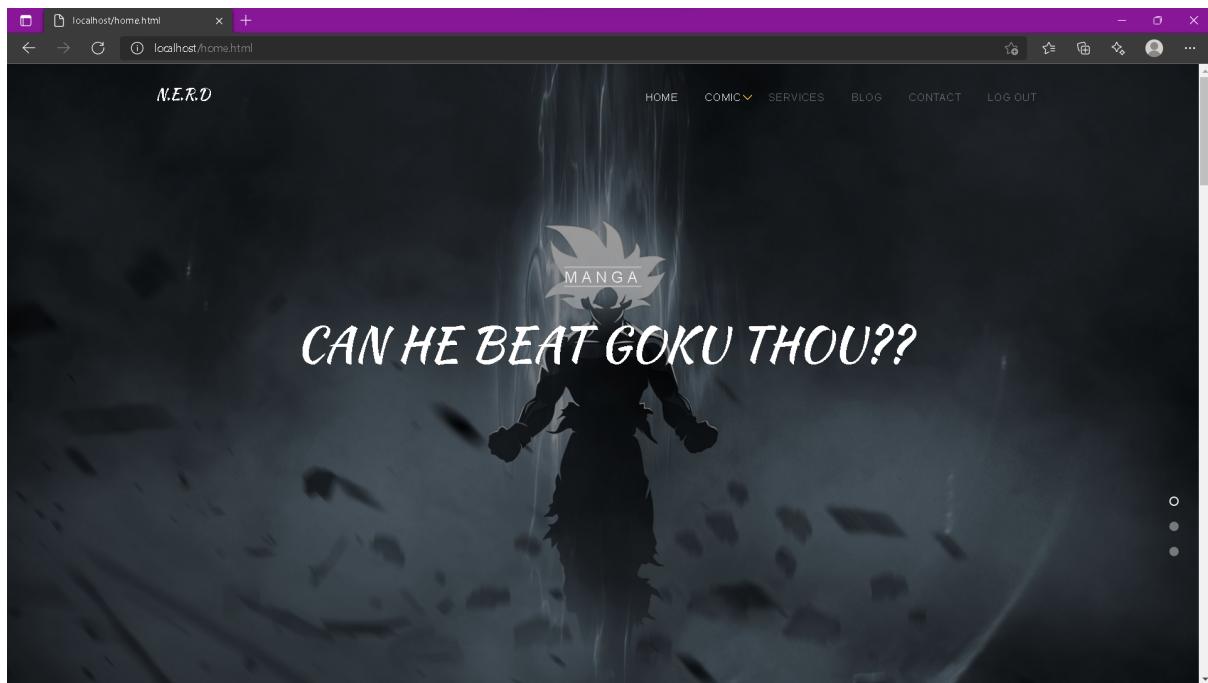
Layer caching:

IMAGE HISTORY			COMMAND
0	COPY . /usr/share/nginx/html # buildkit	340.26 MB	
1	/bin/sh -c #(nop) CMD ["nginx","-g","daemon ...	0 Bytes	
2	/bin/sh -c #(nop) STOPSIGNAL SIGQUIT	0 Bytes	
3	/bin/sh -c #(nop) EXPOSE 80	0 Bytes	
4	/bin/sh -c #(nop) ENTRYPOINT ["/docker-entry...]	0 Bytes	
5	/bin/sh -c #(nop) COPY file:09a214a3e07c919af2fb2d7c749ccbc446b8c10eb217366e5a65640ee9edcc25 in /docker-entrypoint.d	4.61 KB	/bin/sh -c #(nop) COPY file:09a214a3e07c919af2fb2d7c749ccbc446b8c10eb217366e5a65640ee9edcc25 in /docker-entrypoint.d
6	/bin/sh -c #(nop) COPY file:0fd5fca330dcd6a7...	1.04 KB	
7	/bin/sh -c #(nop) COPY file:0b866ff3fc1ef5b0...	1.96 KB	
8	/bin/sh -c #(nop) COPY file:65504f71f5855ca...	1.2 KB	
9	/bin/sh -c set -x && addgroup -g 101 -S nginx ...	17.84 MB	

container:



Container Service exposure:



Docker logs:

```
[C:\Users\luural\Downloads\w3m.com]htmlWEB>docker logs interesting.golick
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: Info: Getting the checksum of /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Starting worker processes... done
/docker-entrypoint.sh: Worker processes are ready for start up
2022/01/28 14:06:59 [notice] 1#1: using the 'epoll' event method
2022/01/28 14:06:59 [notice] 1#1: built by gcc 10.3.1 20210827 (Alpine 10.3.1_git20210827)
2022/01/28 14:06:58 [notice] 1#1: OS: Linux 5.10.16.3-microsoft-standard-WSL2
2022/01/28 14:06:58 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048456
2022/01/28 14:06:58 [notice] 1#1: start worker processes
2022/01/28 14:06:58 [notice] 1#1: start worker process 33
2022/01/28 14:06:58 [notice] 1#1: start worker process 34
2022/01/28 14:06:58 [notice] 1#1: start worker process 35
2022/01/28 14:06:58 [notice] 1#1: start worker process 36
2022/01/28 14:06:58 [notice] 1#1: start worker process 37
2022/01/28 14:06:58 [notice] 1#1: start worker process 38
2022/01/28 14:06:58 [notice] 1#1: start worker process 39
2022/01/28 14:06:58 [notice] 1#1: start worker process 40
2022/01/28 17:0.1 - [28/Jan/2022:14:14:59 +0000] "GET /HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.99 Safari/537.36 Edg/97.0.1072.69"
2022/01/28 17:0.1 - [28/Jan/2022:14:15:00 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://localhost/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.99 Safari/537.36 Edg/97.0.1072.69"
2022/01/28 17:0.1[error]: 200 [open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost", referer: "http://localhost/"]
2022/01/28 17:0.1 - [28/Jan/2022:14:15:13 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://localhost/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.99 Safari/537.36 Edg/97.0.1072.69"
2022/01/28 17:0.1[error]: 200 [open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost", referer: "http://localhost/"]
2022/01/28 17:0.1 - [28/Jan/2022:14:15:14 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.99 Safari/537.36 Edg/97.0.1072.69"
2022/01/28 17:0.1 - [28/Jan/2022:14:15:15 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.99 Safari/537.36 Edg/97.0.1072.69"
2022/01/28 17:0.1 - [28/Jan/2022:14:15:16 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.99 Safari/537.36 Edg/97.0.1072.69"
2022/01/28 17:0.1 - [28/Jan/2022:14:15:17 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.99 Safari/537.36 Edg/97.0.1072.69"
2022/01/28 17:0.1[error]: 200 [open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost", referer: "http://localhost/"]
2022/01/28 17:0.1 - [28/Jan/2022:14:18:19 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://localhost/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.99 Safari/537.36 Edg/97.0.1072.69"
2022/01/28 17:0.1 - [28/Jan/2022:14:18:20 +0000] "GET /home.html HTTP/1.1" 200 15074 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) chrome/97.0.4692.99 Safari/537.36 Edg/97.0.1072.69"
2022/01/28 17:0.1 - [28/Jan/2022:14:18:21 +0000] "GET / HTTP/1.1" 404 555 "http://localhost/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.99 Safari/537.36 Edg/97.0.1072.69"
```

Executing into the container:

```
Rohan@rohan MINGW64 ~/Downloads/web/comic html/WEB DEV (master)
$ docker exec -it interesting_golick sh
/ # ls
bin          media          srv
dev          mnt            sys
docker-entrypoint.d  opt            tmp
docker-entrypoint.sh proc           usr
etc          root           var
home         run
lib
```

Kubernetes:

Starting minikube:

```
C:\Users\mural\Downloads\web\comic html\WEB DEV>minikube start
* minikube v1.25.1 on Microsoft Windows 11 Home 10.0.22000 Build 22000
* Automatically selected the docker driver
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.23.1 preload ...
  > preloaded-images-k8s-v16-v1...: 504.42 MiB / 504.42 MiB 100.00% 45.35 Mi
  > gcr.io/k8s-minikube/kicbase: 0 B [_____] ?% ? p/s 1m15s
* Creating docker container (CPUs=2, Memory=2200MB) ...
* Preparing Kubernetes v1.23.1 on Docker 20.10.12 ...
  - kubelet.housekeeping-interval=5m
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Creating name space:

```
C:\Users\mural\Downloads\web\comic html\WEB DEV>kubectl get namespaces
NAME      STATUS   AGE
default   Active   85m
kube-node-lease   Active   85m
kube-public   Active   85m
kube-system   Active   85m

C:\Users\mural\Downloads\web\comic html\WEB DEV>kubectl apply -f ns.yaml
namespace/comic created

C:\Users\mural\Downloads\web\comic html\WEB DEV>kubectl get namespaces
NAME      STATUS   AGE
comic     Active   61s
default   Active   92m
kube-node-lease   Active   92m
kube-public   Active   92m
kube-system   Active   92m

C:\Users\mural\Downloads\web\comic html\WEB DEV>
```

```
! ns.yaml
1  apiVersion: v1
2  kind: Namespace
3  metadata:
4    name: comic
```

Creating pod:

```

! deploymentyaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    labels:
5      app: comic
6    name: comic-deployment
7    namespace: comic
8  spec:
9    replicas: 1
10   selector:
11     matchLabels:
12       app: comic
13   template:
14     metadata:
15       labels:
16         app: comic
17     spec:
18       containers:
19         - image: nerd:v1
20           imagePullPolicy: Never
21           name: comic
22           ports:
23             - containerPort: 80

```

Pod deployment:

```

C:\Users\mural\Downloads\web\comic html\WEB DEV>kubectl get deployments -n comic
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
comic-deployment      1/1     1            1           99s

C:\Users\mural\Downloads\web\comic html\WEB DEV>kubectl get pods -n comic
NAME                  READY   STATUS    RESTARTS   AGE
comic-deployment-7886f9fd6b-j4fbt   1/1     Running   0          2m52s

```

Configuring deploying pod with 2 replicas:

```
C:\Users\mural\Downloads\web\comic html\WEB DEV>kubectl apply -f deployment.yaml
deployment.apps/comic-deployment configured

C:\Users\mural\Downloads\web\comic html\WEB DEV>kubectl get pods -n comic
NAME                  READY   STATUS    RESTARTS   AGE
comic-deployment-7886f9fd6b-cxb9t   1/1     Running   0          32s
comic-deployment-7886f9fd6b-j4fbt   1/1     Running   0          6m18s
```

Service Exposure:

```
! service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: ccomic-service
5    namespace: comic
6  spec:
7    selector:
8      app: comic
9    type: LoadBalancer
10   ports:
11     - protocol: TCP
12       port: 8080
13       targetPort: 80
14       nodePort: 30000
```

```
C:\Users\mural\Downloads\web\comic.html\WEB DEV>kubectl get service -n comic
NAME        TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
ccomic-service   LoadBalancer   10.96.78.218   <pending>      8080:30000/TCP   5s

C:\Users\mural\Downloads\web\comic.html\WEB DEV>kubectl get service -n comic
NAME        TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
ccomic-service   LoadBalancer   10.96.78.218   <pending>      8080:30000/TCP   93s

C:\Users\mural\Downloads\web\comic.html\WEB DEV>kubectl apply -f service.yaml
The Service "ccomic-service" is invalid: spec.ports[0].nodePort: Invalid value: 30000: provided port is already allocated

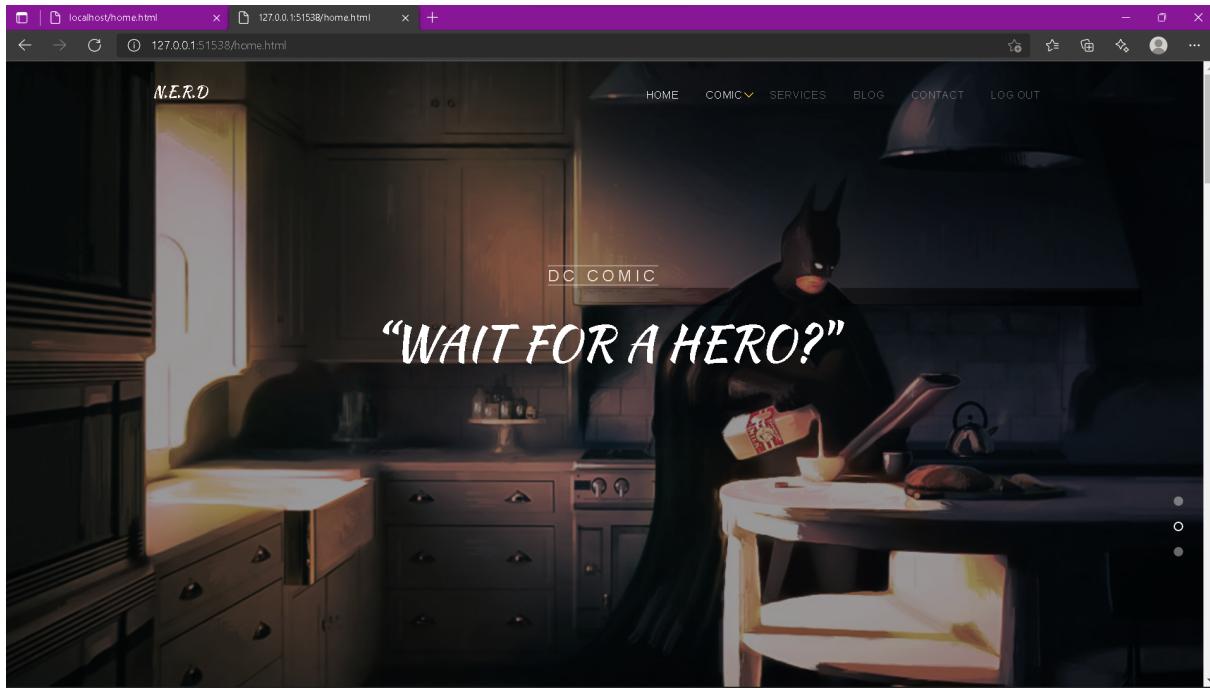
C:\Users\mural\Downloads\web\comic.html\WEB DEV>kubectl get service -n comic
NAME        TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
ccomic-service   LoadBalancer   10.96.78.218   <pending>      8080:30000/TCP   115s

C:\Users\mural\Downloads\web\comic.html\WEB DEV>minikube service cccomic-service -n comic
| Executing "docker container inspect minikube --format=[{{.State.Status}}]" took an unusually long time: 2.155205s
* Restarting the docker service may improve performance.

X Exiting due to SVC_NOT_FOUND: Service 'cccomic-service' was not found in 'comic' namespace.
You may select another namespace by using 'minikube service cccomic-service -n <namespace>'. Or list out all the services using 'minikube service list'
```

```
C:\Users\mural\Downloads\web\comic.html\WEB DEV>minikube service cccomic-service -n comic
|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|
| comic | cccomic-service | 8080 | http://192.168.49.2:30000 |
|-----|
* Starting tunnel for service cccomic-service.
|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|
| comic | cccomic-service | 8080 | http://127.0.0.1:51538 |
|-----|
* Opening service comic/cccomic-service in default browser...
| Because you are using a Docker driver on Windows, the terminal needs to be open to run it.
```

Hosted Webpage:

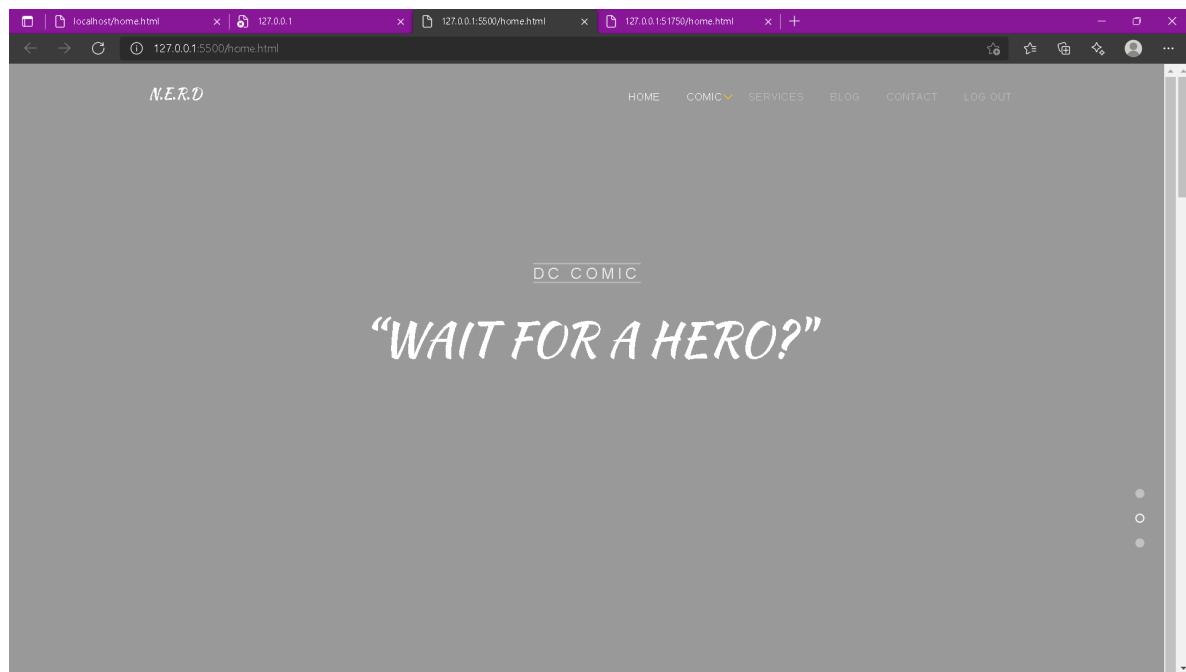


Resource limitation:

```
C:\Users\mural\Downloads\web\comic.html\WEB DEV>kubectl describe pod comic-deployment-65775ddc4-wk4fx -n comic
Name:           comic-deployment-65775ddc4-wk4fx
Namespace:      comic
Priority:       0
Node:          minikube/192.168.49.2
Start Time:    Fri, 28 Jan 2022 21:41:23 +0530
Labels:         app=comic
                pod-template-hash=65775ddc4
Annotations:   <none>
Status:        Pending
IP:            172.17.0.5
IPs:
  IP:          172.17.0.5
Controlled By: ReplicaSet/comic-deployment-65775ddc4
Containers:
  comic:
    Container ID:          nerd:v1
    Image:                 nerd:v1
    Image ID:               sha256:...
    Port:                  80/TCP
    Host Port:             0/TCP
    State:                 Waiting
      Reason:               CreateContainerError
    Ready:                 False
    Restart Count:          0
    Limits:
      cpu:        100m
      memory:    1Mi
    Requests:
      cpu:        100m
      memory:    1Mi
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-vrhrh (ro)
Conditions:
  Type        Status
  Initialized  True
  Ready       False
  ContainersReady  False
  PodScheduled  True
Volumes:
  kube-api-access-vrhrh:
    Type:           Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:   kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI:    true
  QoS Class:      Guaranteed
  Node-Selectors: <none>
  Tolerations:    node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
```

ROLLBACK STRATEGY

Version-2



Rolling back to version 1:

```
C:\Users\mural\Downloads\web\comic html\WEB DEV>kubectl rollout history deploy comic-deployment -n comic
deployment.apps/comic-deployment
REVISION  CHANGE-CAUSE
1          <none>
2          <none>
3          <none>

C:\Users\mural\Downloads\web\comic html\WEB DEV>kubectl rollout undo deploy comic-deployment --to-revision=2 -n comic
deployment.apps/comic-deployment rolled back

C:\Users\mural\Downloads\web\comic html\WEB DEV>kubectl rollout undo deploy comic-deployment --to-revision=1 -n comic
deployment.apps/comic-deployment rolled back

C:\Users\mural\Downloads\web\comic html\WEB DEV>minikube service ccomic-service -n comic
! Executing "docker container inspect minikube --format={{.State.Status}}" took an unusually long time: 2.4570468s
* Restarting the docker service may improve performance.

|-----|-----|-----|-----|
| NAMESPACE |     NAME      | TARGET PORT |          URL           |
|-----|-----|-----|-----|
|   comic    | ccomic-service |        8080 | http://192.168.49.2:30000 |
|-----|-----|-----|-----|
* Starting tunnel for service ccomic-service.

|-----|-----|-----|-----|
| NAMESPACE |     NAME      | TARGET PORT |          URL           |
|-----|-----|-----|-----|
|   comic    | ccomic-service |            | http://127.0.0.1:62710 |
|-----|-----|-----|-----|
* Opening service comic/ccomic-service in default browser...
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```

